FERMILAB-SLIDES-20-109-SCD

G4Opticks for Liquid Argon TPC's

Outline

- Motivation: physics of liquid Argon TPC's.
- Opticks/G4Opticks
- G4OpticksTest
- Progress so far
- Work plan - To do list
- Summary

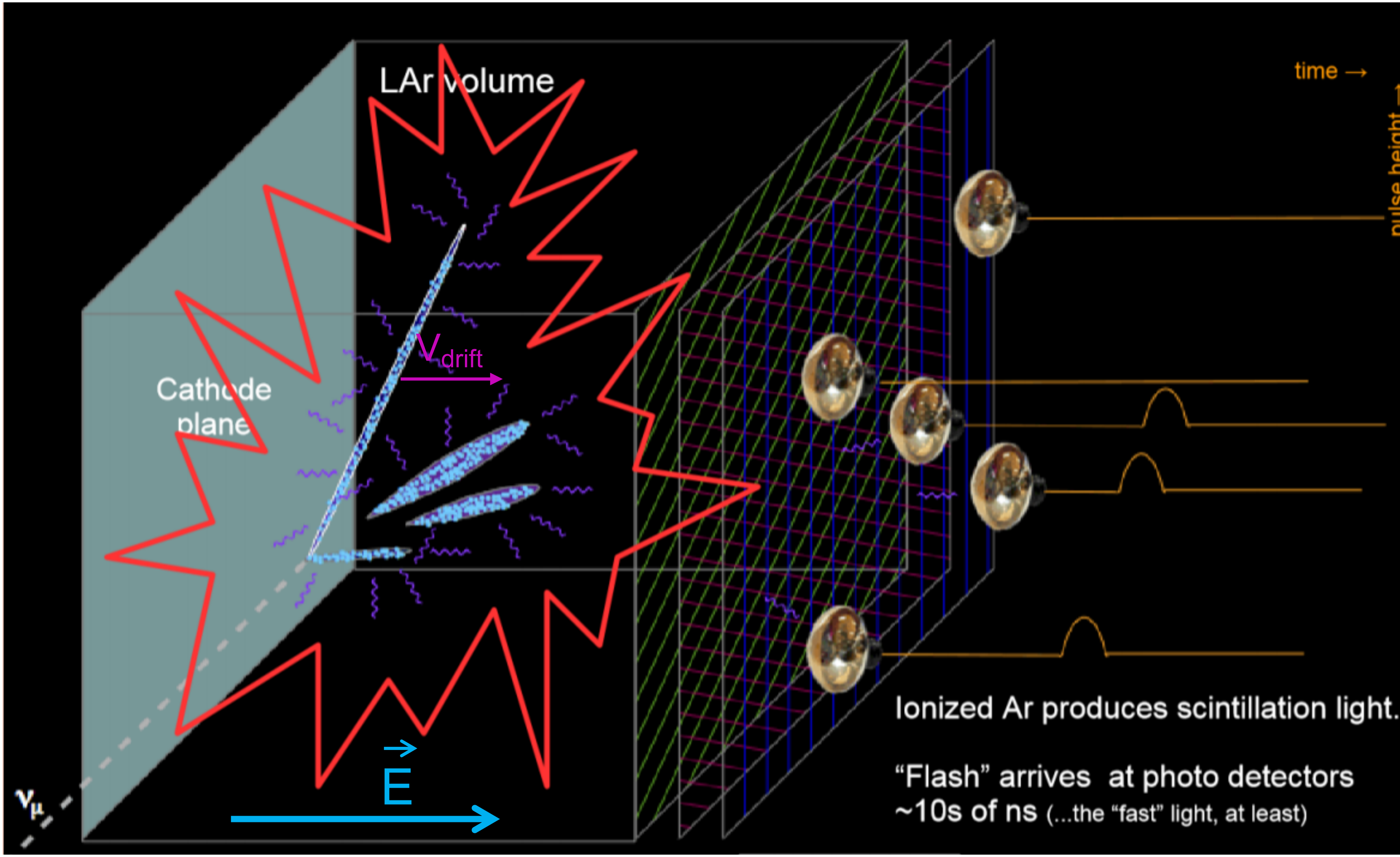
Hans Wenzel[1]
Krzysztof Genser[1]
Soon Yung Jun[1]
Alexei Strelchenko[1]

[1)] Fermilab

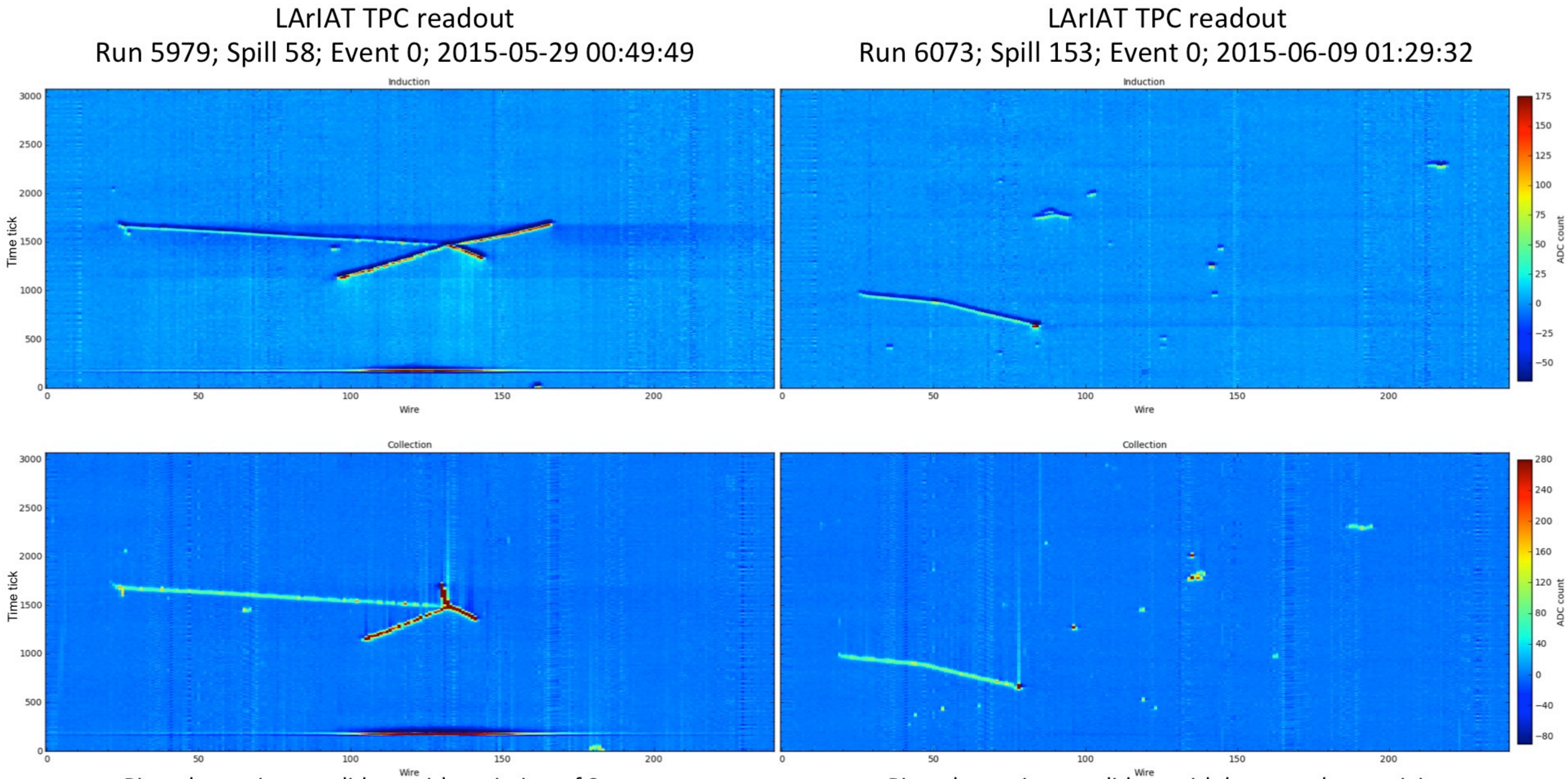
Special thanks to Simon Blyth!

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

Single-phase liquid Argon TPC



Single-phase liquid Argon TPC



Pion absorption candidate with emission of 3 protons

Pion absorption candidate with large nuclear activity

Scintillation/Ionization properties in liquid Argon

Light yield ~ few 10,000's of photons per MeV (dependences on E field, particle type and purity)
where minimum ionization is 2.105 MeV/cm

Wavelength of emission is 128nm

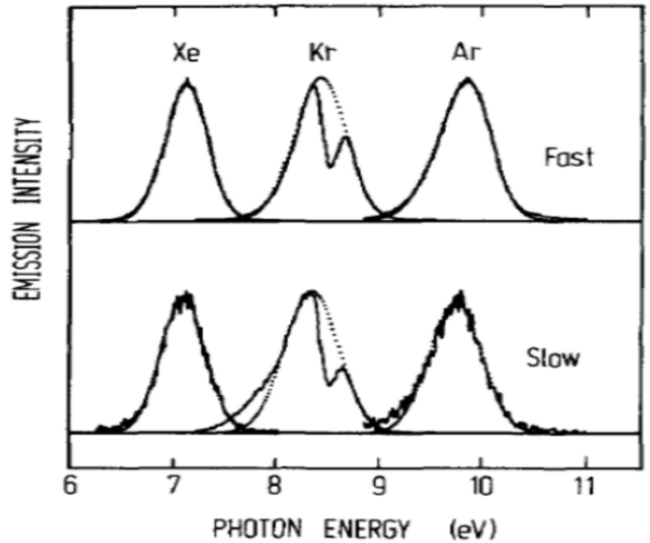
Light with two characteristic time constants:

- *fast component, 6 ns*
- *slow component, 1500 ns*

Argon is highly transparent to its own scintillation light.

Rayleigh scattering length ~ 50-60cm.

J Chem Phys vol 91 (1989) 1469 E Morikawa et al



Presence of electric field → allows for separation of ions and electrons created in Ionization process:

- electrons drift to read out plane and produce signal.
- slow moving Ions might contribute to space charge effects.
- ionization and scintillation light are 'anti'- correlated in complicated way.
- photons traced by Opticks.
- electrons are collected in Hit collection and then traced by separate module.

Motivation

- Using Geant4 to simulate photon propagation on the CPU takes ~hours to simulate 1 event for a typical liquid Argon TPC (lArTPC).

- currently lArTPC experiments use Look Up Tables (LUT) or parameterizations for photon response. But:
  - only approximation,
  - LUT grow with detector size to a point that jobs can't run on a typical grid node,
  - still need full simulation to create the tables to run on traditional grids.

- Simon Blyth showed that Opticks speeds up the photon simulation to the level where it is as fast as the rest of the simulation
  - allows to run full optical simulation event by event,
  - allows to investigate various ideas for improvements like:
    - Improve calorimetric energy resolution,
    - Using ratios like fast/slow scintillation light or light/ionization for Particle ID,
    - ….
- While liquid Argon TPCs are a main priority for the laboratory many experiments can benefit e.g. dark matter searches, dual readout calorimeters (e.g. Crystals: effect of Cerenkov light directionality, different TPC configurations …), various groups are investigating Opticks → we want to develop a flexible framework (like artg4tk see: https://cdcvs.fnal.gov/redmine/projects/artg4tk/wiki/Artg4tk).

Opticks

https://simoncblyth.bitbucket.io/env/presentation/opticks_may2020_hsf.html

EPJ Web of Conferences **214**, 02027 (2019)
https://doi.org/10.1051/epjconf/201921402027
Opticks : GPU Optical Photon Simulation for Particle Physics using NVIDIA® OptiX™
Simon Blyth

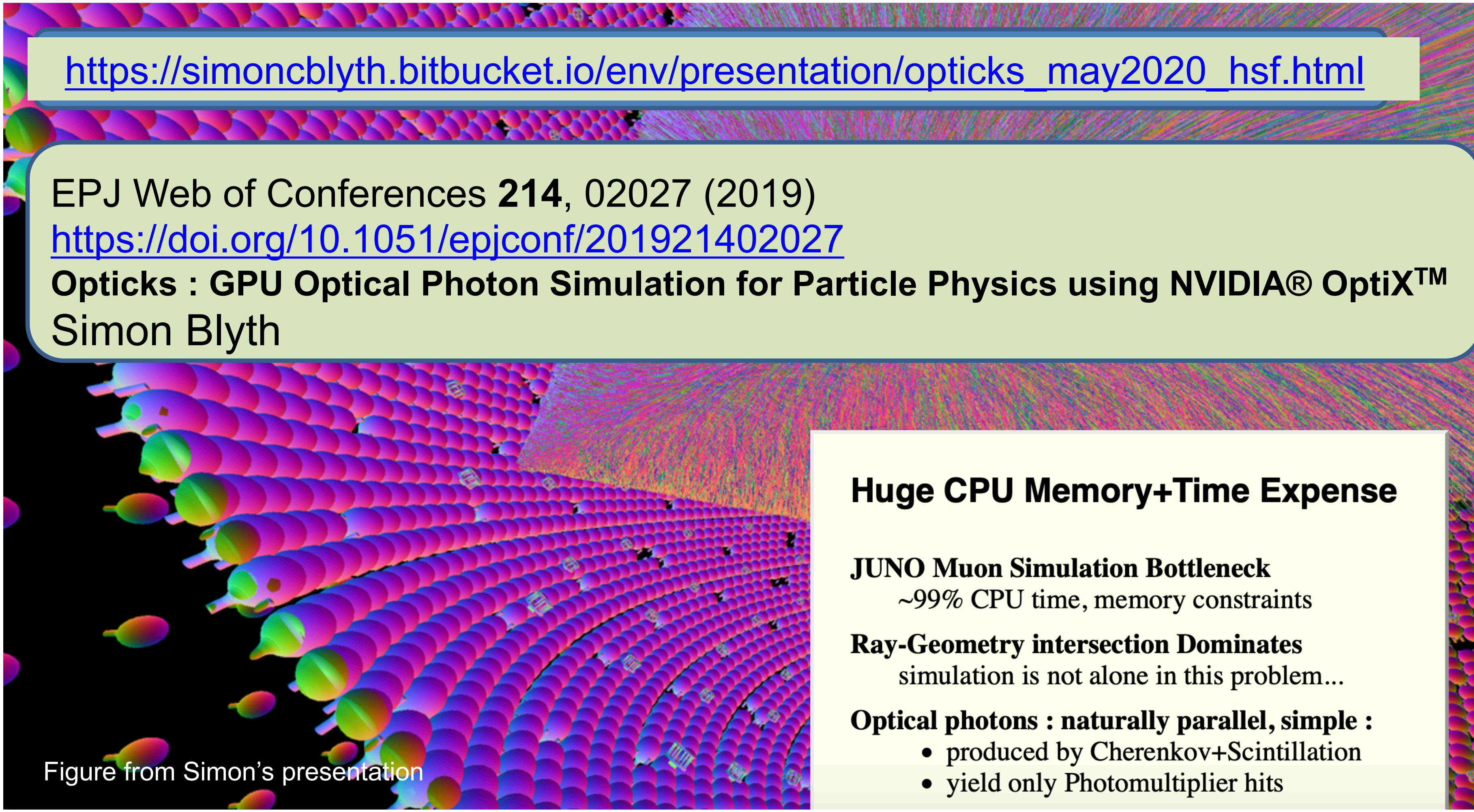


Figure from Simon's presentation

Opticks

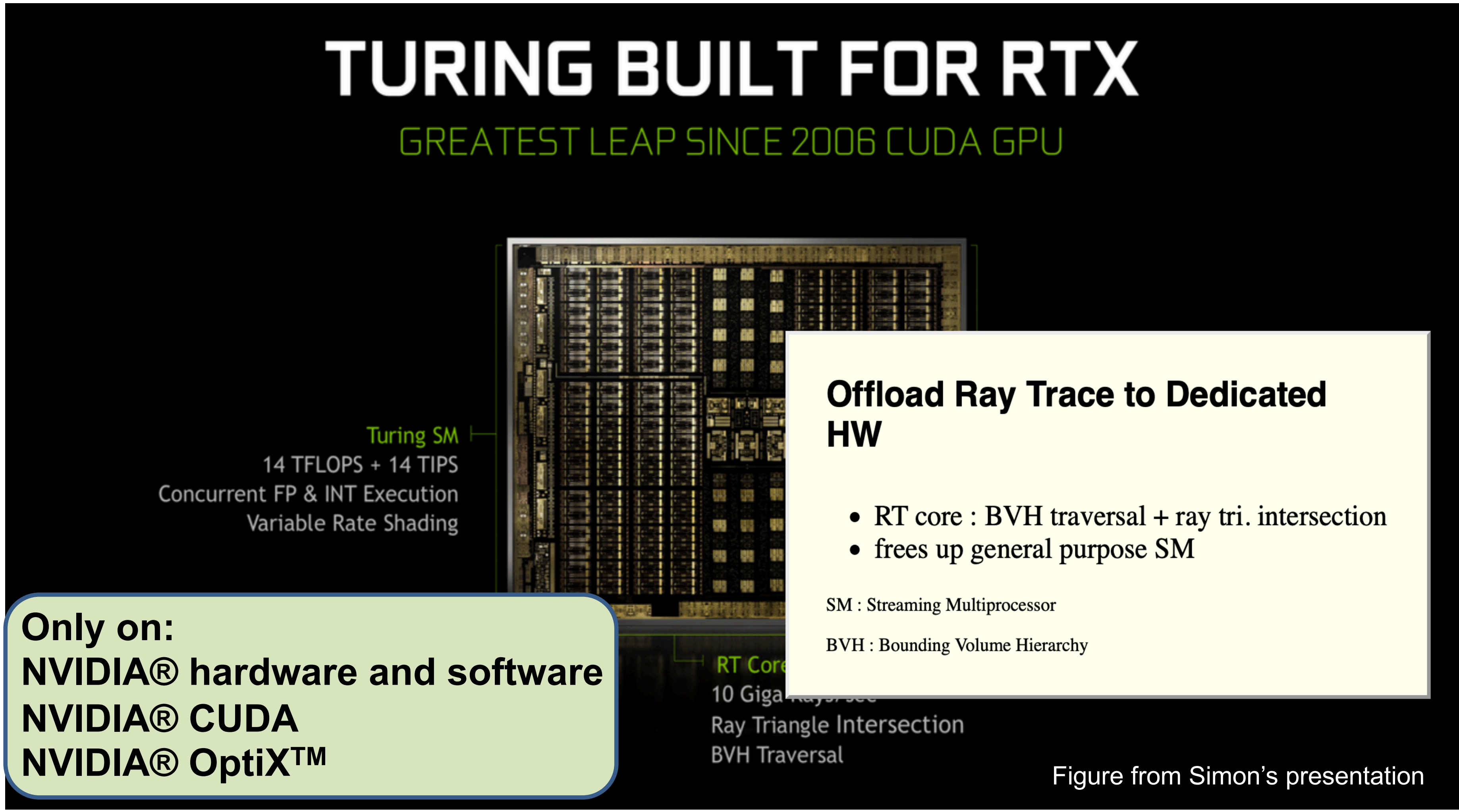


Figure from Simon's presentation

Only on:
NVIDIA® hardware and software
NVIDIA® CUDA
NVIDIA® OptiX™

Opticks

Open source: https://bitbucket.org/simoncblyth/opticks/

Opticks : Translates G4 Optical Physics to CUDA/OptiX

OptiX : single-ray programming model -> line-by-line translation

CUDA Ports of Geant4 classes

- G4Cerenkov (only generation loop)
- G4Scintillation (only generation loop)
- G4OpAbsorption
- G4OpRayleigh
- G4OpBoundaryProcess (only a few surface types)

Modify Cherenkov + Scintillation Processes

- collect *genstep*, copy to GPU for generation
- avoids copying millions of photons to GPU

Scintillator Reemission

- fraction of bulk absorbed "reborn" within same thread
- wavelength generated by reemission texture lookup

Opticks (OptiX/Thrust GPU interoperation)

- **OptiX** : upload gensteps
- **Thrust** : seeding, distribute genstep indices to photons
- **OptiX** : launch photon generation and propagation
- **Thrust** : pullback photons that hit PMTs
- **Thrust** : index photon step sequences (optional)

GPU Resident Photons

Seeded on GPU

associate photons -> *gensteps* (via seed buffer)

Generated on GPU, using genstep param:

- number of photons to generate
- start/end position of step

Propagated on GPU

Only photons hitting PMTs copied to CPU

Thrust: **high level C++ access to CUDA**

thrust

Code at the speed of light

- https://developer.nvidia.com/Thrust

Figure from Simon's presentation

Need: wave-length shifting process on GPU

G4Opticks hybrid workflow

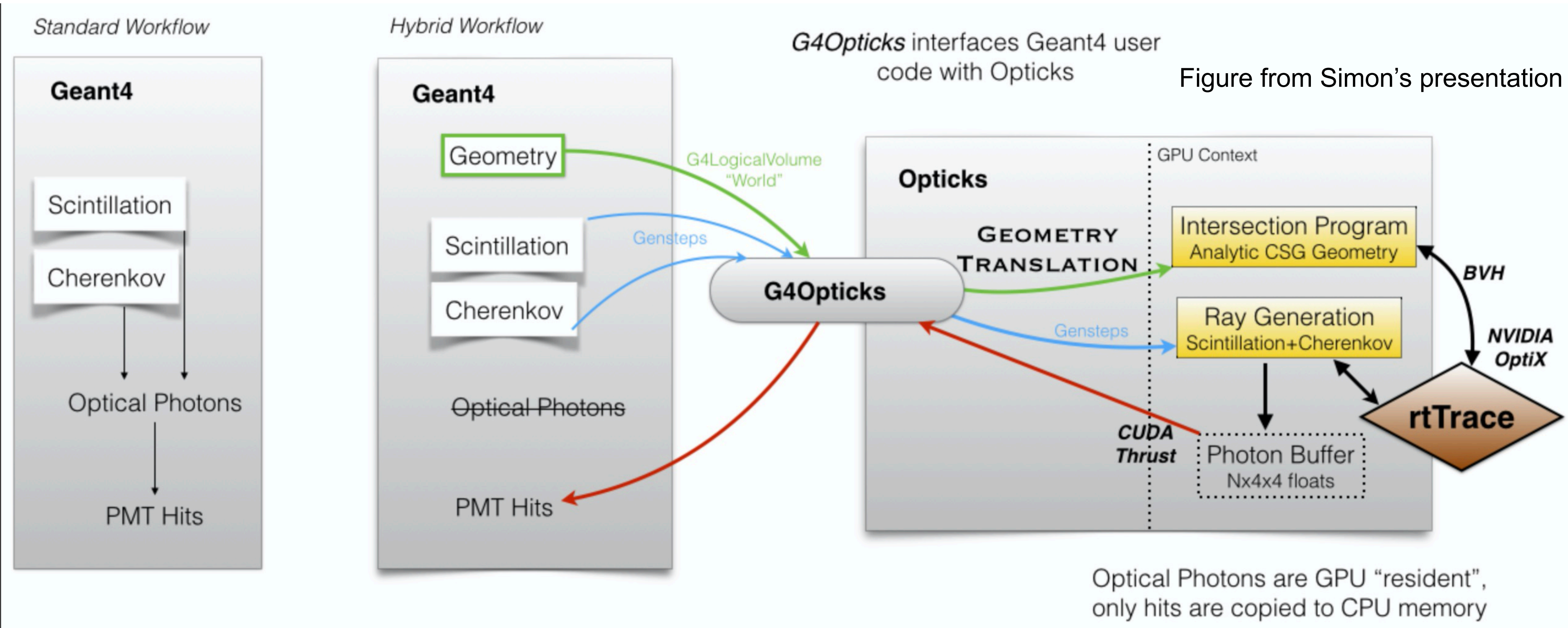


Figure from Simon's presentation

Plans with respect to evolving G4Opticks:

- Move the harvesting of Cerenkov and Scintillation Gensteps functionality to UserSteppingAction/Sensitive detector and make use of Geant4 API's.
- Use the same implementation of the scintillation process on CPU and GPU, use the same optical properties.
- Make optical photon processing concurrent with the rest of the event→ use G4Tasking by J. Madsen.

Gensteps

A Genstep collects all information necessary to generate Scintillation and Cerenkov photons on the GPU.

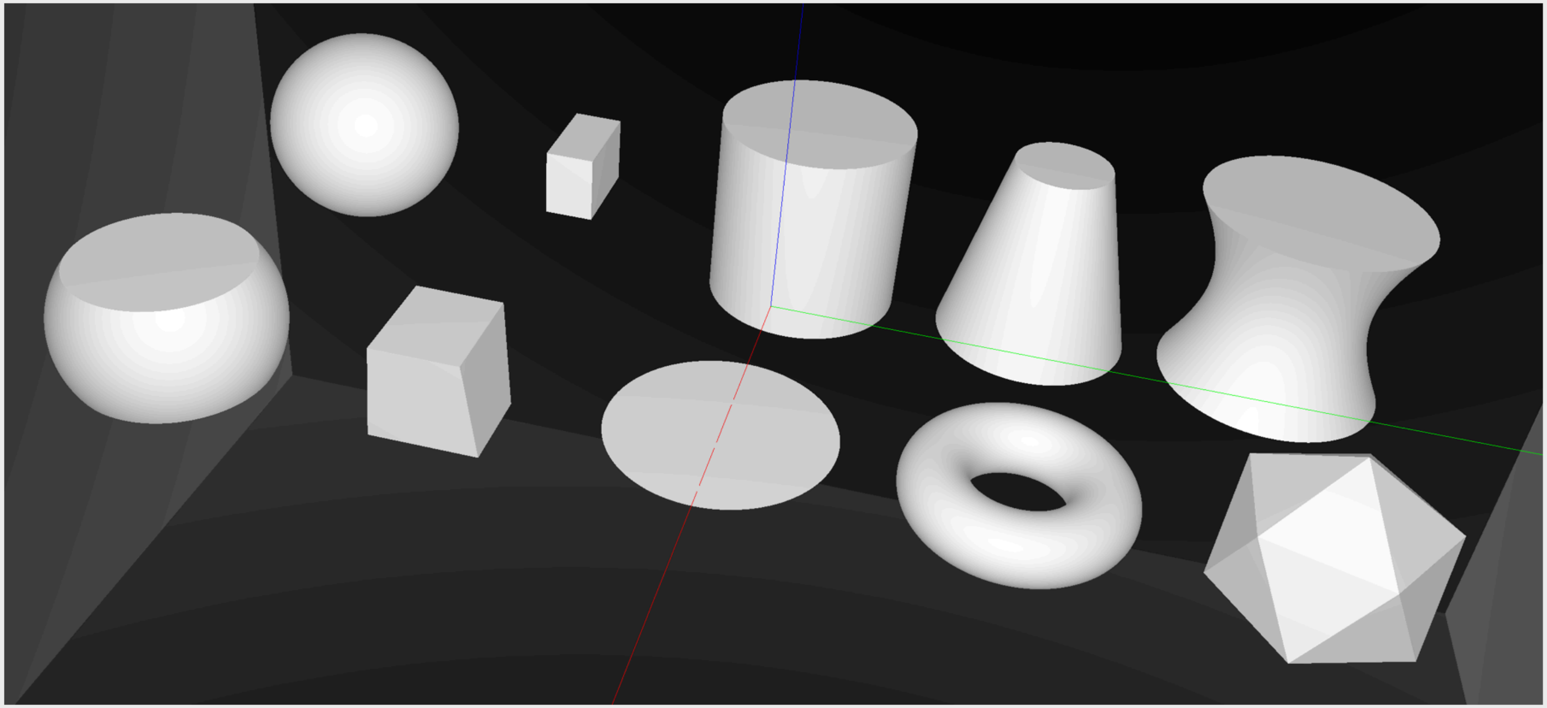
Scintillation Genstep:

```
    G4StepPoint* pPreStepPoint = aStep->GetPreStepPoint();
//   G4StepPoint* pPostStepPoint = aStep->GetPostStepPoint();
    G4ThreeVector x0 = pPreStepPoint->GetPosition();
    G4ThreeVector p0 = aStep->GetDeltaPosition().unit();
    G4double t0 = pPreStepPoint->GetGlobalTime();
    if (photons > 0) {
        G4Opticks::GetOpticks()->collectScintillationStep(
                //1, // 0    id:zero means use scintillation step count
                OpticksGenstep_G4Scintillation_1042,
                aTrack->GetTrackID(),
                materialIndex,
                photons,
                x0.x(), // 1
                x0.y(),
                x0.z(),
                t0,
                deltaPosition.x(), // 2
                deltaPosition.y(),
                deltaPosition.z(),
                aStep->GetStepLength(),
                definition->GetPDGEncoding(), // 3
                definition->GetPDGCharge(),
                aTrack->GetWeight(),
                pPreStepPoint->GetVelocity(),
                scntId,
                YieldRatio, // slowerRatio,
                FastTimeConstant, // TimeConstant,
                SlowTimeConstant, //slowerTimeConstant,
                ScintillationTime, //scintillationTime,
                0.0, //wrong but not used scintillationIntegrationMax,
                0, //spare1
                0 // spare2
                );
    }
```

G4Solid -> CUDA Intersect Functions for ~10 Primitives

- 3D parametric ray : **ray(x,y,z;t) = rayOrigin + t * rayDirection**
- implicit equation of primitive : **f(x,y,z) = 0**
- -> polynomial in **t** , roots: **t > t_min** -> intersection positions + surface normals

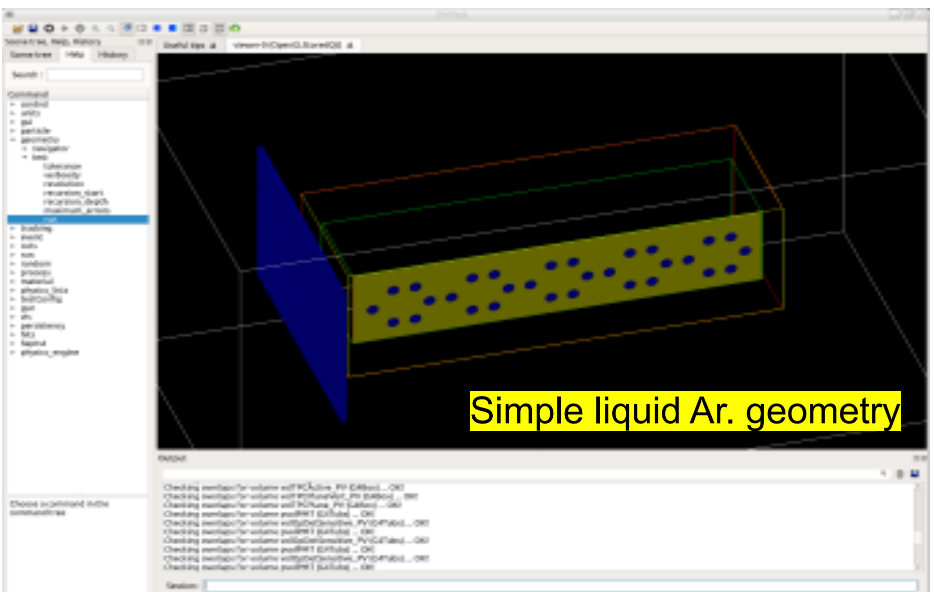
Figure from Simon's presentation



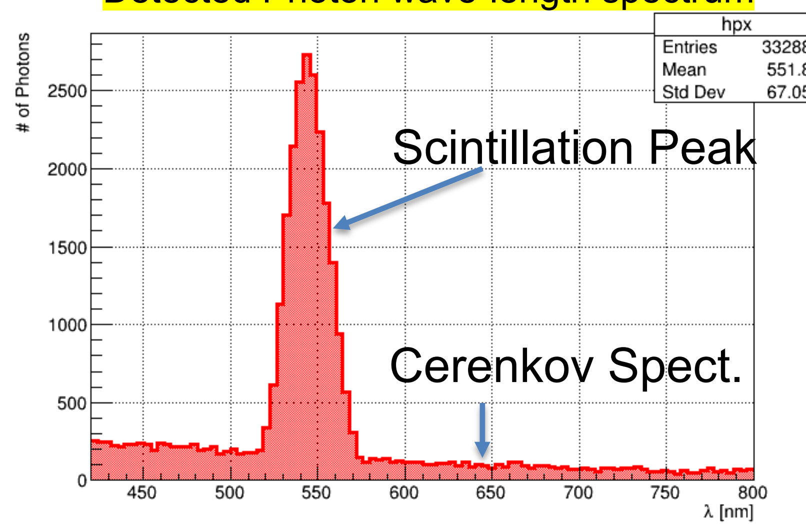
Sphere, Cylinder, Disc, Cone, Convex Polyhedron, Hyperboloid, Torus, ...

Progress so far

- Got CerenkovMinimal working. CerenkovMinimal is an application provided by Opticks that demonstrates the use of G4Opticks.
- Implemented the harvesting of scintillation Gensteps; now part of Opticks.
- Modified and documented build procedure (e.g. on FNAL GPU servers) especially using preexisting external libraries and newer versions thereof; fed back to Opticks.
- Moved to geant4.10.6.p02 → allows use of e.g. G4PhysListFactoryAlt, Optical physics constructor, latest API's.
- Wrote G4OpticksTest: a Geant4 application making use of G4Opticks, with emphasis on liquid Argon TPC's; see next slide.
- We use our own fork of Opticks: https://github.com/hanswenzel/opticks → pull requests.



Detected Photon wave-length spectrum



G4OpticksTest

G4OpticksTest (in progress) : Geant4 example that demonstrates the use of the G4Opticks hybrid workflow:

https://github.com/hanswenzel/G4OpticksTest

Features are based on experience with artg4tk:

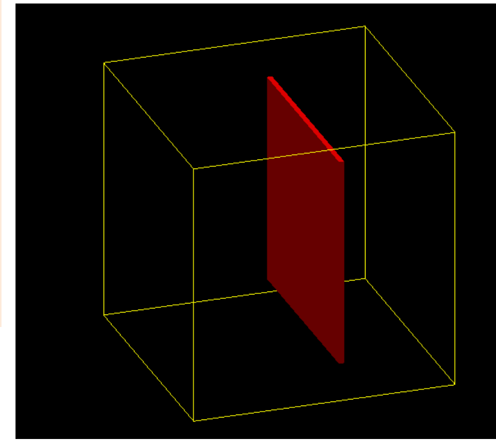
- Uses Geant4 to harvest Scintillation and Cerenkov Gensteps. The harvesting is moved to sensitive Detectors.
- Uses Opticks to generate and propagate optical photons.
- Uses gdml with extensions for flexible Detector construction and provide optical properties. gdml extensions include:
  - assigning sensitive detector to logical Volumes. A library of various sensitive detector types is provided (specifically lArTPCSD).
  - assigning step-limits to logical Volume to match Geant4 steps and TPC readout pitch.
  - assigning visualization properties.
  - assigning homogenous electric field.
- Uses G4PhysListFactoryAlt (R. Hatcher) to define and configure physics using reference physics lists, electromagnetic options and physics constructors, e.g.:

G4PhysListRegistry::GetModularPhysicsList <FTFP_BERT+OPTICAL+STEPLIMIT+NEUTRONLIMIT>, as "FTFP_BERT" with extensions "+OPTICAL+STEPLIMIT+NEUTRONLIMIT”

Performance: very preliminary!

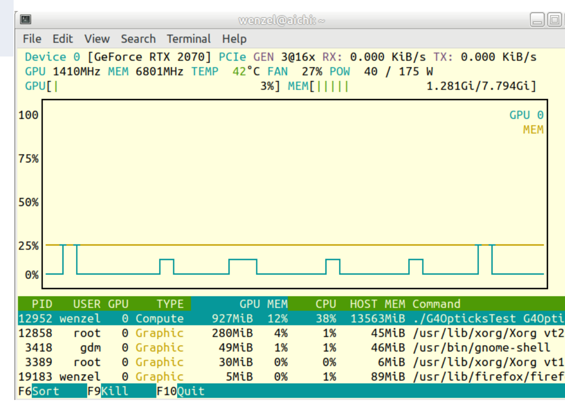
Hardware:	
CPU	Intel(R) Core(TM) i7-9700K 3.60GHz
GPU	GeForce RTX 2070" CUDA Driver Version /11.0 CUDA Capability: 7.5 VRAM: 7981 Mbytes Cores: 2304

Geometry:
water: 1x1 m^2
embedded glass detector
photon yield 500/MeV
single 1GeV muon



	real	user	sys
Geant4	39m14.560s	20m24.484s	18m0.377s
Geant4+Opticks	3m10.631s	1m22.088s	38.289s

In this configuration we don't make use of all the GPU resources. In liquid Argon more than 100x as many optical photons are produced. Very simple geometry.
Sequential: Geant4-Opticks, Geant4 takes most of the time in this case.



Work plan-To do list

- Create an extended Geant4 example with Opticks
  - Provide persistency for Hit collection.
  - Provide timing/memory use results.
  - Implement current Geant4 Scintillation process on GPU.
  - Implement wavelength shifting process on GPU.
  - Specifically provide realistic Geant4 stand-alone liquid Argon TPC example.
  - Provide Sensitive detector plugins for different detector types.

- Packaging, make it available to community.
  - Provide docker/singularity image
    - Might involve integration with artg4tk.

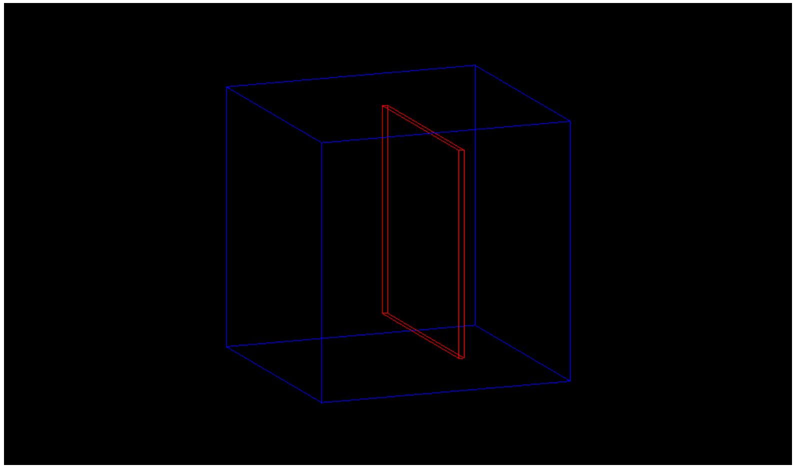
- Use G4Tasking (by J. Madsen) for true concurrency.
  - Install and test Opticks with the latest Geant4 reference releases and a lArTPC geometry.
  - Develop Geant4 Task application with Opticks where:
    - Gensteps chunks are collected in-situ during the tracking/stepping loop,
    - once a predetermined chunk size is reached the optical photon propagation is offloaded to the GPU (device), while the rest of simulation and Gensteps collection continues on the CPU (host).

Summary

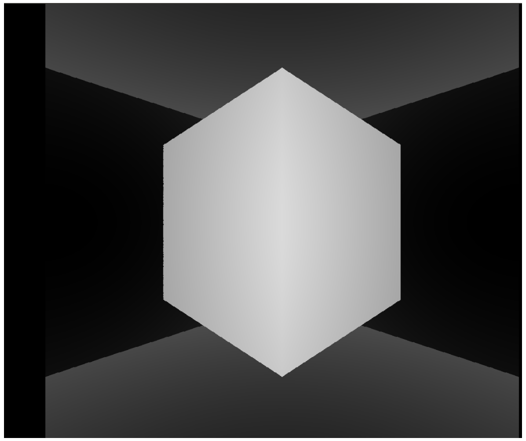
- We have ported Simon Blyth's Opticks to Geant4 10.6.p02 and fed back the required changes.
- A first preliminary look at timing results looks very promising.
- We are working on an extended liquid Argon TPC example where we use G4tasking to dispatch processing of the optical photons to the GPUs using Opticks while processing the rest of the event on the CPU.

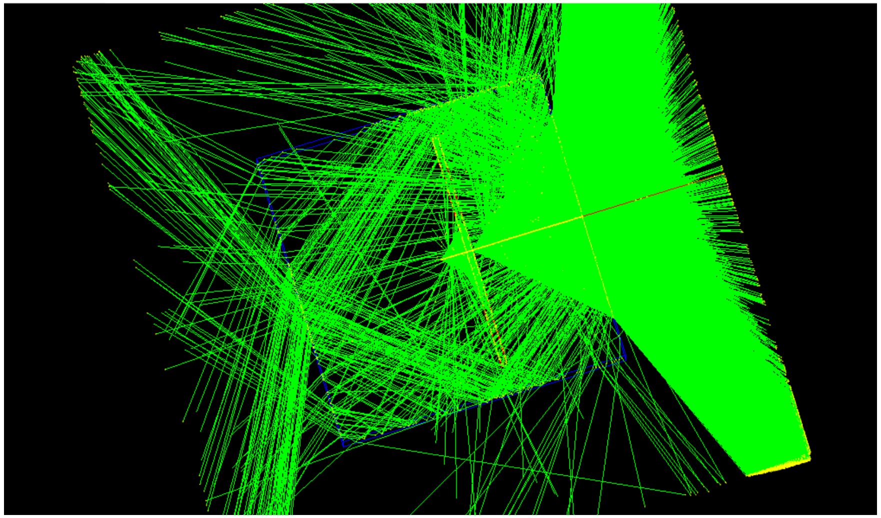
Backup Slides

CerenkovMinimal

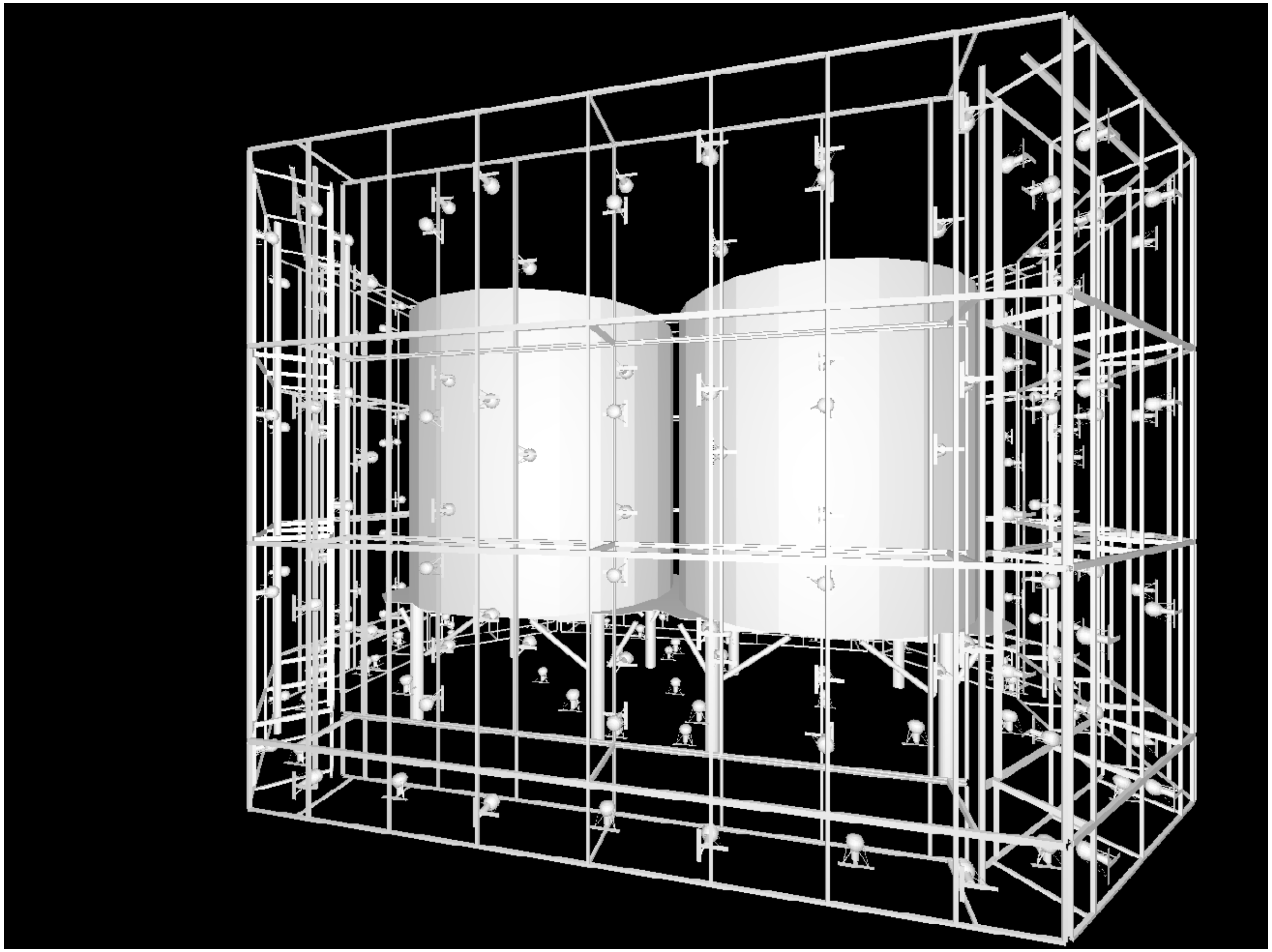


Opticks CerenkovMinimal to create:
Geocache as well as gdml and gltf (opengl) output files.
visualize the geometry with opengl.
The gdml file was then modified to define sensitive Volumes and then fed into larTest for a full Geant 4 simulation of optical photons to get some timing results.

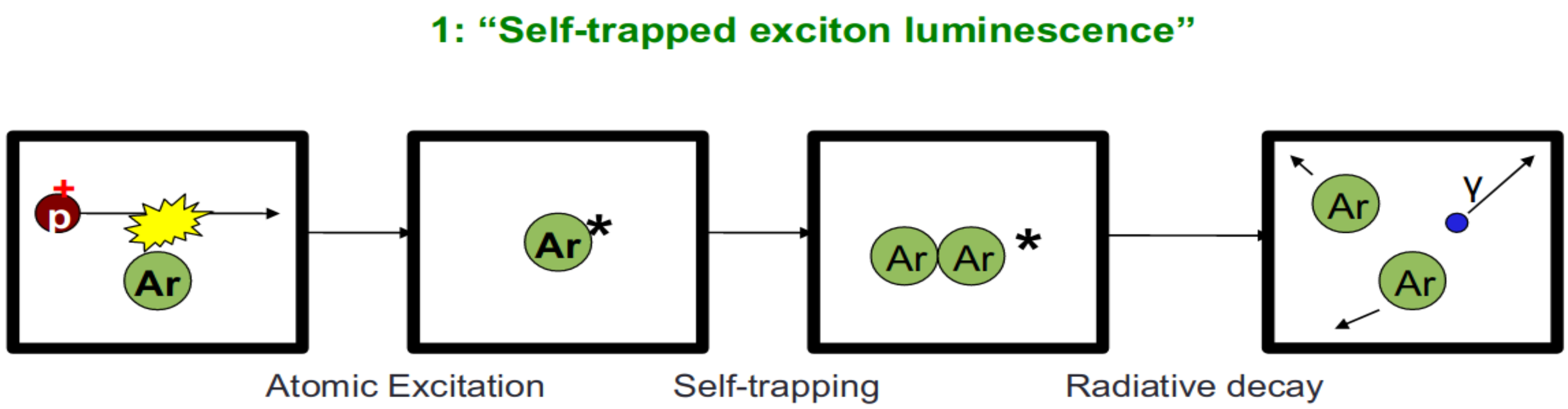




Visualization of Daya Bay Detector

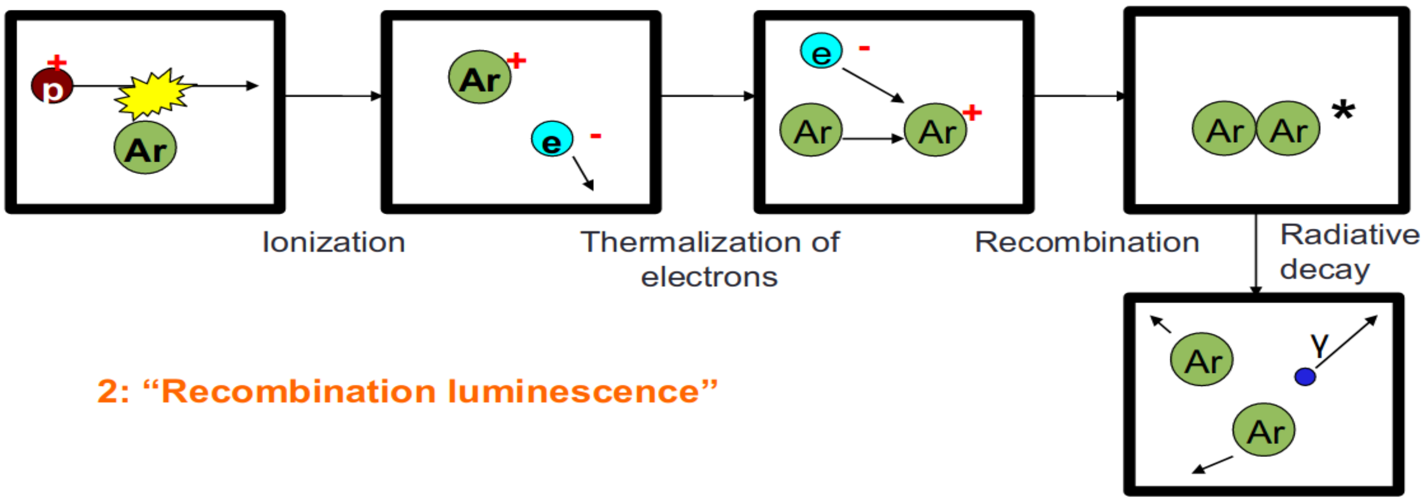


Mechanisms of Scintillation in LAr



From: INTRODUCTION TO SCINTILLATION LIGHT INLIQUID ARGON Ben Jones, MIT
https://microboone-exp.fnal.gov/public/talks/LArTPCWorkshopScintLight_bjpjone_2014.pdf

Mechanisms of Scintillation in LAr



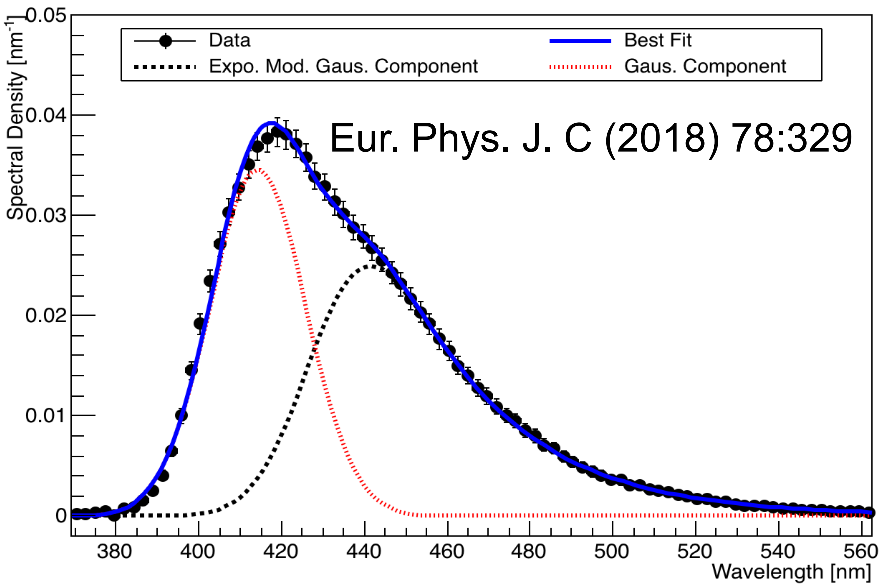
2: "Recombination luminescence"

Recombination step involves an electron cloud around the track core

-> E-Field dependent scintillation yield
-> dE/dx dependent scintillation yield
-> Charge and light anti-correlation

From: INTRODUCTION TO SCINTILLATION LIGHT INLIQUID ARGON Ben Jones, MIT
https://microboone-exp.fnal.gov/public/talks/LArTPCWorkshopScintLight_bjpjone_2014.pdf

Argon scintillates in the vacuum ultraviolet spectrum, which requires wavelength shifting to convert the VUV photons to visible photons so they can be detected.
The wavelength shifter used is 1,1,4,4-tetraphenyl-1,3-butadiene (TPB). It has a fast re-emission time on the order of 1 ns and a high conversion efficiency for VUV light.
TPB can be applied to reflecting liner of the TPC (LarIAT) and to input window of photo-detector.
The wave length shifting process is not implemented in Opticks yet, though the similar scintillation re-emission process is.
Wave length shifting changes energy, direction, polarization and delays time of photon.



Status: Available installations

	Lq cluster	laptop	desktop
hardware	Tesla V100	Geforce MX150	GeForce RTX2070
cores	5120	384	2304
VRAM	32 Gb	2Gb	8Gb
OS	Sl7.7	Ubuntu 20.04	Ubuntu 18.04
Driver	450.36.06	450.51.05	450.51.05
gcc	8.3.0	9.3.0	7.5.0
Optix	6.5.0	6.5.0	6.5.0
CUDA	10.1	11.0	11.0
Cuda Capability	7.0	6.1	7.5

Geant4 version used geant4.10.06.p02

artg4tk/larg4